

Page 11, bridging page 12, delete the last paragraph and insert the new last paragraph as follows:

C\

After comparing the test code specified in the addressed Treg byte with each of the condition codes CCX0 ... CCX7 (assuming the operation is being executed on the X side of the machine), then the specified operation is carried out on the SIMD lanes where there is a match, and is not carried out on the SIMD lanes where there is no match. An example is illustrated in Figure 7. Assumed that the operation illustrated in Figure 6 and described above has been carried out and that condition codes CCX0 to CCX7 have been set as described above depending on the results of the arithmetic operation in each of the SIMD lanes $b_0 \dots b_7$. It is assumed for this example that the condition codes are b_0 0010, b_1 0101, b_2 0011, b_3 0010, b_4 0010 b_5 0100. This is illustrated in the condition code register in Figure 7. Let us also assume that the addressed test register byte in the TST field of the instruction holds the condition code 0011. This denotes the condition Carry Set C. SIMD satisfies this condition. Assume that the subsequent operation to be carried out is also an ADD instruction operating on the byte packed contents of two source registers SRC1, SRC2 with the results to be loaded into a destination register DST. Because a test register byte has been specified, the addition operation is only effected on the SIMD lanes where the condition code set for that lane (CCX0 ... CCX7) satisfies the condition defined by the test code set in the addressed Treg byte. This is determined by a condition code checker 50. The output of the condition code checker 50 controls a set of switches 52, one for each SIMD lane $b_0 \dots b_7$. These switches control whether or not the results of the addition operation recited in the instruction update the values in the corresponding lane in the destination register DST. This is shown diagrammatically in Figure 7, with a cross illustrating that the result of the addition operation does not get loaded into the destination

c register, and a through arrow illustrating that it does. This is denoted in the destination register by DST denoting an original byte in the destination register (that is prior to execution of the instruction), and RES denoting a result byte which, following execution of the instruction is a result of the arithmetic operation on that lane.
